PATENT
Docket No. **SYB/0093.01**

IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

In re application of: Rajesh Chawla et al.	Examiner: Lovel, Kimberly M
Serial No.: 10/707,471	Art Unit: 2167
Filed: December 16, 2003	APPEAL BRIEF
For: System with Methodology for Executing Relational Operations Over Relational Data and Data Retrieved from SOAP Operations	

Mail Stop Appeal
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Sir:

BRIEF ON BEHALF OF RAJESH CHAWLA

This is an appeal from the Final Rejection mailed June 14, 2007, in which currently-pending claims 1-55 stand finally rejected. Appellant filed a Notice of Appeal on September 17, 2007. This brief is submitted electronically in support of Appellant's appeal.

TABLE OF CONTENTS

1. REAL PARTY IN INTEREST 3

2. RELATED APPEALS AND INTERFERENCES 3

3. STATUS OF CLAIMS 3

4. STATUS OF AMENDMENTS 3

5. SUMMARY OF CLAIMED SUBJECT MATTER 4

6. GROUNDS OF REJECTION TO BE REVIEWED 7

7. ARGUMENT 7

A. First Ground: Claims 1-55 rejected under 35 U.S.C. 103(a) 7

8. CLAIMS APPENDIX 16

9. EVIDENCE APPENDIX 24

10. RELATED PROCEEDINGS APPENDIX 25

1. REAL PARTY IN INTEREST

The real party in interest is assignee Sybase, Inc. located at One Sybase Drive, Dublin, CA 94568.

2. RELATED APPEALS AND INTERFERENCES

There are no appeals or interferences known to Appellant, the Appellant's legal representative, or assignee which will directly affect or be directly affected by or have a bearing on the Board's decision in the pending appeal.

3. STATUS OF CLAIMS

The status of all claims in the proceeding is as follows:

Rejected: Claims 1-55

Allowed or Confirmed: None

Withdrawn: None

Objected to: None

Canceled: None

Identification of claims that are being appealed: Claims 1-55

An appendix setting forth the claims involved in the appeal is included as Section 8 of this brief.

4. STATUS OF AMENDMENTS

Two Amendments have been filed in this case. Appellant mailed an Amendment on October 3, 2006 in response to a non-final Office Action dated May 3, 2006. In response to a non-final Office Action dated December 28, 2006, Appellant filed an Amendment on March 28, 2007. In the Amendment filed on March 28, 2007, the pending claims were amended in a manner which Appellant believes clearly distinguished the claimed invention over the art of record, for overcoming the art rejections. In response to the Examiner's Final Rejection dated June 14, 2007 (hereinafter "Final Rejection") finally rejecting Appellant's claims, Appellant filed a Notice of

Appeal. Appellant also filed a Request for Reconsideration on October 15, 2007 requesting reconsideration of the Final Rejection. This Request for Reconsideration did not amend any of the claims, but rather requested reconsideration of the prior art rejections made by the Examiner in the Final Rejection. In response to the Request for Reconsideration, the Examiner issued an Advisory Action dated October 29, 2007 which maintained the rejection of Appellant's claims. Accordingly, no amendments to the claims have been entered in this case after the date of the Final Rejection.

5. SUMMARY OF CLAIMED SUBJECT MATTER

As to Appellant's **First Ground** for appeal, Appellant asserts that the art rejection of Appellant's claims 1-55 under 35 USC Section 103(a) relying on the combination of US PGPub 2003/0093436 to Brown et al (hereinafter "Brown") and US PGPub 2005/0044164 to O'Farrell et al (hereinafter "O'Farrell") fails to teach or suggest all of the claim limitations of Appellant's rejected claims 1-55, where the claimed invention is set forth in the embodiment in **independent claim 1:** A method for performing database operations on data obtained from a web service (Appellant's specification, paragraph [0014], paragraph [0056], paragraph [0059]; also see generally, Fig. 4 and Figs. 5A-B), the method comprising: creating at least one proxy table in a database, each proxy table mapping to a method of the web service (Appellant's specification, paragraph [0014], paragraph [0059], paragraph [0073], paragraphs [0075]-[0077], paragraph [0091]; Fig. 5A at 502; see also paragraphs [0113]-[0117]), generating meta data about the mapping and storing the meta data in a database table of the database (Appellant's specification, paragraph [0014], paragraph [0080], paragraph [0085], paragraph [0090]; Fig. 4 at 422, 430; see also paragraphs [0113]-[0117]), in response to a database operation on a particular proxy table, using the meta data for converting the database operation into a format for invoking a particular method of the web service based upon the corresponding mapping (Appellant's specification, paragraph [0014], paragraph [0077], paragraph [0080], paragraphs [0092]-[0094]; Fig. 5A at 503, 504, 505), invoking the particular method of the web service (Appellant's specification, paragraph [0014], paragraph [0080], paragraphs [0086]-[0087], paragraph [0095]; Fig. 5A at 506; also see paragraphs

[0119]-[0130]), converting results obtained from invoking the particular method into data for use at the database based upon the corresponding mapping (Appellant's specification, paragraph [0014], paragraph [0081], paragraph [0095]; Fig. 5B at 507; also see paragraph [0124] and paragraphs [0131]-[0134]), performing the database operation on the data at the database to generate a result set (Appellant's specification, paragraph [0014], paragraphs [0058]-[0059], paragraph [0074], paragraph [0081], paragraph [0096]; Fig. 5B at 508; also see paragraph [135]), and returning the result set in response to the database operation (Appellant's specification, paragraph [0014], paragraph [0058], paragraph [0081], paragraph [0096]; Fig. 5B at 509).

For Appellant's argument under the **First Ground** for appeal, Appellant additionally argues that the art rejection under **Section 103(a)** relying on the combination of Brown and O'Farrell fails to teach or suggest all of the claim limitations of Appellant's rejected claims, where the claimed invention is set forth in the embodiment in **independent claim 22:** In a computer connected to a network and having access to a remote service (Appellant's specification paragraph [0015], paragraph [0060], paragraph [0074]; see generally, Fig. 4), a system for performing operations at a database on data obtained from the remote service (Appellant's specification paragraph [0015], paragraphs [0058]-[0059], paragraph [0074], paragraph [0081], paragraph [0096]; Fig. 5B at 508-509), the system comprising: a mapping module for creating database tables representing at least some methods of the remote service accessed through a defined interface and storing mapping data regarding methods of the remote service in a database system table (Appellant's specification, paragraph [0015], paragraph [0073], paragraph [0075], paragraph [0080], paragraph [0085], paragraph [0091], paragraph [0094]; Fig. 4 at 422, 430; Fig. 5A at 501-502; see also paragraphs [0113]-[0117]), an invocation module for converting a database operation on a database table representing a method of the remote service into a call for invoking the method using the mapping data (Appellant's specification, paragraph [0015], paragraph [0080], paragraph [0087], paragraph [0095]; Fig. 4 at 420; Fig. 5A at 503-505; also see paragraphs [0119]-[0130]), a communication module for transmitting the call for invoking the method to the remote service, and returning result values from invoking the method to the database (Appellant's specification, paragraph [0015], paragraphs [0080]-[0081], paragraphs [0086]-[0087],

paragraph [0095]; Fig. 4 at 423; Fig. 5A at 506; Fig. 5B at 507; also see paragraphs [0121]-[0131]), and a conversion module for converting result values received from the method into database format, performing the database operation on the converted result values to generate a database result set, and returning the database result set in response to the database operation (Appellant's specification, paragraph [0015], paragraphs [0058]-[0059], paragraph [0074], paragraph [0081], paragraph [0095]; Fig. 5B at 507-509; also see paragraphs [0131]-[0135]).

For Appellant's argument under the **First Ground** for appeal, Appellant additionally argues that the art rejection under **Section 103(a)** relying on the combination of Brown and O'Farrell fails to teach or suggest all of the claim limitations of Appellant's rejected claims, where the claimed invention is set forth in the embodiment in **independent claim 40:** In a database system, a method for performing database queries on data available from an application (Appellant's specification, paragraph [0016], paragraphs [0056]-[0060], paragraphs [0079]-[0081]; also see generally, Fig. 4 and Figs. 5A-B and paragraphs [0090]-[0096]), the method comprising: establishing communication between a database and an application having an interface (Appellant's specification, paragraph [0016], paragraph [0061], paragraph [0073], paragraph [0090], Fig. 4 at 420, Fig. 5A at 501), creating database tables to represent at least some functions of the application based on the interface, each database table mapping to a corresponding function of the application (Appellant's specification, paragraph [0016], paragraph [0059], paragraph [0073], paragraphs [0075]-[0077], paragraph [0091]; Fig. 5A at 502; see also paragraphs [0113]-[0117]), generating meta data about the mapping and storing the meta data in a system table of the database (Appellant's specification, paragraph [0016], paragraph [0080], paragraph [0085], paragraph [0090]; Fig. 4 at 422, 430; see also paragraphs [0113]-[0117]), in response to a database query received on a database table corresponding to a function of the application, generating input arguments expected by the function based on the database query and the mapping meta data (Appellant's specification, paragraph [0014], paragraph [0077], paragraph [0080], paragraphs [0092]-[0094]; Fig. 5A at 503, 504, 505), invoking the function with the input arguments and receiving results from invoking the function (Appellant's specification, paragraph [0016], paragraph [0080], paragraphs [0086]-[0087], paragraph [0095]; Fig. 5A at 506; also see

paragraphs [0119]-[0130]), converting the results into a database result set (Appellant's specification, paragraph [0016], paragraph [0081], paragraph [0095]; Fig. 5B at 507; also see paragraph [0124] and paragraphs [131]-[134]), and returning the database result set in response to the database query (Appellant's specification, paragraph [0016], paragraph [0058], paragraph [0081], paragraph [0096]; Fig. 5B at 508-509).

6. GROUNDS OF REJECTION TO BE REVIEWED

The grounds for appeal are:

(1st) Whether claims 1-55 are unpatentable under 35 U.S.C. 103(a) as being obvious over US PGPub 2003/0093436 to Brown et al ("Brown") in view of US PGPub 2005/0044164 to O'Farrell et al ("O'Farrell").

7. ARGUMENT

A. First Ground: Claims 1-55 rejected under 35 U.S.C. 103(a)

1. General

Under Section 103(a), a patent may not be obtained if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which the subject matter pertains. To establish a prima facie case of obviousness under this section, the Examiner must establish: (1) that there is some suggestion or motivation, either in the references themselves or in the knowledge generally available to one of ordinary skill in the art, to modify the reference or to combine reference teachings, (2) that there is a reasonable expectation of success, and (3) that the prior art reference (or references when combined) must teach or suggest all the claim limitations. (See e.g., MPEP 2142). The reference(s) cited by the Examiner fail to meet these conditions.

2. Claims 1-55

The Examiner has rejected claims 1-55 under 35 U.S.C. 103(a) as being obvious over US PGPub 2003/0093436 to Brown et al ("Brown") in view of US PGPub 2005/0044164 to O'Farrell et al ("O'Farrell"). The following rejection of Appellant's claim 1 by the Examiner is representative of the Examiner's rejection of the Appellant's

claims as being unpatentable over Brown and O'Farrell:

> Referring to claim 1, Brown discloses a method for performing database operations on data obtained from a web service, the method comprising:
> creating at least one proxy table in a database, each proxy table mapping to a method of the web service [creating a virtual table representative of the web service] (Brown: see [0062]-[0063] and [0074]);
> in response to a database operation on a particular proxy table, converting the database operation into a format for invoking a particular method of the web service based upon the corresponding mapping (Brown: see [0049]);
> invoking the particular method of the web service (Brown: see [0057]-[0059]);
> converting results obtained from invoking the particular method into data for use at the database based upon the corresponding mapping (Brown: see [0074]); and
> performing the database operation on the data at the database to generate a result set (Brown: see [0075]-[0077], lines 1-2); and
> returning the result set in response to the database operation (Brown: see [0075}-[0077], lines 1-2).
>
> However, Brown fails to explicitly disclose the further limitations of generating meta data about the mapping and storing the meta data in a database table of the database and using the meta data for converting the database operation into a format for invoking a particular method of the web service based upon the corresponding mapping. O'Farrell discloses using web services to retrieve data from multiple enterprise data stores (see [0012]), including the further limitations of generating meta data [metadata 312] about the mapping and storing the meta data in a database table of the database (see [0074], lines 8-12 and Fig 3) and using the meta data for converting the database operation into a format for invoking a particular method of the web service based upon the corresponding mapping (see [0076]).
>
> It would have been obvious to one of ordinary skill in the art at the time the invention was made to utilize the mapping structure of O'Farrell with the method of Brown by replacing the DADX files with the mapping structure. One would have been motivated to do so in order to provide a form of automation, which yields significant savings and efficiencies (O'Farrell: see [0005]).
>
> (Final Rejection, paragraph 6, pages 3-4)

Appellant's claimed invention is distinguishable from Brown and O'Farrell in a number of respects. Appellant's invention creates mappings to methods of Web services and encapsulates these mappings in proxy tables that are used to represent methods of Web services (Appellant's specification, paragraphs [0128]-[0129]). During the creation of these proxy tables, meta data about these mappings is automatically generated and

stored by Appellant's system to enable the remote Web service to be located and called in response to an operation on the proxy tables (Appellant's specification, paragraph [0091]). Significantly, Appellant's system automatically creates the proxy table and related mappings given a Web Services Description Language file describing the Web service (Appellant's specification, paragraph [0077] and paragraph [0091]). Appellant's system stores the mapping meta data in system tables of the database (Appellant's specification, paragraph [0085]). This meta data is used when a database operation (e.g., SQL SELECT operation) on the proxy table representing the remote Web service is received to map the relational data types to the appropriate representation expected by the Web method (Appellant's specification, paragraph [0091] and paragraph [0094]). These features are specifically included as limitations of Appellant's claims. For example, Appellant's Claim 1 includes the following claim limitations:

> A method for performing database operations on data obtained from a web service, the method comprising:
> creating at least one proxy table in a database, each proxy table mapping to a method of the web service;
> generating meta data about the mapping and storing the meta data in a database table of the database;
> in response to a database operation on a particular proxy table, using the meta data for converting the database operation into a format for invoking a particular method of the web service based upon the corresponding mapping;
> invoking the particular method of the web service;
> converting results obtained from invoking the particular method into data for use at the database based upon the corresponding mapping;
> performing the database operation on the data at the database to generate a result set; and
> returning the result set in response to the database operation.
>
> (Appellant's claim 1, emphasis added)

In contrast to Appellant's invention, Brown's system relies on mapping information stored in a file that is external to the database which is referred to as "DADX" or "Document Access Definition" file (Brown, paragraph [0045]). This is also shown at Fig. 6 which illustrates the DADX file 51 associated with the Websphere Application Server 63 (Brown, Fig. 6). The DADX file is a configuration file that defines the operations that can be performed by the Web Service (Brown, paragraph

[0050]). As described at paragraph [0029] of Brown:

> The service provider creates DAD and DADx documents and deploys them to the web application. Each DADx document is associated with a URL that identifies a specific web service.
>
> (Brown, paragraph [0029], emphasis added)

Significantly, the DADX file is a user-specified mapping file that creates the associations between the relational data and XML document structure. It is not automatically generated by Brown's system. This reliance on input of the mapping file is described by Brown as follows:

> One of the inputs into both storage and retrieval is the user-specified mapping file 37 that creates the association between relational data and XML document structure. This mapping file 37 is called a Document Access Definition (DAD) 37 and provides a way to create an XML document 35 and specify the XML elements, attributes and shape desired. The focus of this approach is in moving and manipulating XML documents.
>
> (Brown, paragraph [0045]), emphasis added)

As described above, Brown's system relies on a user-specified mapping file which is received as input. Appellant's invention, in contrast, automatically creates the proxy table and generates mapping meta data which is stored for subsequent use when operations on proxy tables representing the remote service are performed. Thus, Appellant's solution automates the process of integrating a remote service and does not rely on input of a mapping to the remote service. In addition, Appellant's solution stores data about the mappings to methods of the remote service in database system tables, which provides several benefits compared to Brown's approach of relying on external mapping files. A primary benefit is that storing the mapping data in the database means that existing database backup functions can be used for backing up this data. Replication is also easier because the meta data is stored in the database and, as a result, standard database replication methods may be used.

In the Final Rejection the Examiner acknowledges that Brown fails to disclose the limitations of generating meta data about the mappings, storing the meta data in a database table of the database, and using the meta data for invoking a particular method

of the Web service based on the corresponding mappings. Therefore, the Examiner adds O'Farrell as providing these teachings. However, when one reviews the O'Farrell reference, one finds that its teachings are not comparable to Appellant's claimed invention as will be next discussed.

Although O'Farrell discusses meta data, O'Farrell makes no mention of using the meta data to invoke a method of a Web service. In O'Farrell's system, the meta data specifies how data from different enterprise data sources will be stored and related in a mobile client which is connected to one or more backend data stores through a middle tier application server (O'Farrell, paragraphs [0071]-[0074]; Fig. 3). More particularly, the Examiner references paragraph [0076] of O'Farrell as providing the teaching of using meta data for converting a database operation into a format for invoking a particular method of a Web service. However, paragraph [0076] of O'Farrell provides as follows:

> In the metadata 312, the data definition from the enterprise data sources is mapped to views that are used to create the data store on the client and store the relevant business data on the mobile client from the enterprise data sources in a relational database. Access to this business data is performed via a the business object layer defined and stored in metadata on the mobile client. As shown in FIG. 3, the ORDER_ID from the ERP data source is mapped to a business object property called OrderID, whos relational definition is stored in metadata 318 on the mobile client 316 and utilized by one or more the mobile applications also defined in metadata. The F_NAME data from the CRM enterprise data source is mapped to (stored into) the FirstName business object property definition stored in the mobile client database, and the L_NAME data is mapped to the LastName business object property. Similarly, the CRED_LIM data from the HR/Finance data source is mapped to the CreditLimit business object property, and the WARRANTY data from the Legacy/ODBC data source is mapped to the Warranty business object propery. Thus, data from the potentially dissimilar and incompatible disparate enterprise data sources 302, 304, 306, 308, 310 are delivered to the mobile client through the Data Manager Web Services to the local data store (represented by the lines from the enterprise data sources to the application server 314) in the proper format for access using one of the business objects on the mobile client (indicated in the mobile client 316 with actual values).
>
> (O'Farrell, paragraph [0076], emphasis added)

As illustrated above, in O'Farrell's system, the mapping is from data definitions from enterprise data sources to views that are used to create data stores on the client

(O'Farrell, paragraphs [0075]-[0076]). In other words, the mapping described in O'Farrell is a mapping between data fields of a client device and data fields of one or more back end data sources. O'Farrell provides no teachings of using the meta data for purposes of invoking a method of a Web service. To the extent O'Farrell describes a Web service, it describes a "Connector Web Service" which acts an intermediary between a client and an enterprise data source. Additionally, O'Farrell's system does not operate in response to a database operation, nor does it convert the database operation into a format for invoking a method of a Web service as provided, for example, in the following limitations of Appellant's claim1:

> in response to a database operation on a particular proxy table, using the meta data for converting the database operation into a format for invoking a particular method of the web service based upon the corresponding mapping;
>
> (Appellant's claim 1, emphasis added)

Another difference between Appellant's invention and the Brown and O'Farrell solutions is that Appellant's invention automatically creates proxy tables and generates related mapping meta data which is stored for subsequent use when operations on the proxy tables representing the Web service are performed. Thus, Appellant's solution automates the process of integrating a remote Web service and does not rely on input of a mapping to the remote service or require user configuration of the mapping. As previously described, Brown's system does not include this feature as it relies on a user-specified DADX file which is received as input. O'Farrell does not cure these deficiencies of Brown as it relies on a user creating and configuring views so as to map enterprise data source information with business objects on the client device (O'Farrell, paragraphs [0050]-[0055]). O'Farrell's system includes a "Studio" component that can be used to configure the system, including the mapping meta data (O'Farrell, paragraph [0036]; Fig. 1 at 110, 112, 130). However, this tool does not automatically generate mappings to methods of a Web service. Rather, it relies on a developer or user to define mappings between fields on the client devices and those of enterprise data sources (O'Farrell, paragraphs [0052]-[0054]). Thus, its teachings are not at all comparable to those of Appellant's claimed invention.

Additionally, O'Farrell's system maintains the mapping meta data at a middle tier application server which acts as an intermediary between a client and one or more data sources (e.g., Dextera Server 314 as shown at Fig. 3 of O'Farrell). Appellant's claimed invention does not require any such middle tier or intermediary. As previously described, Appellant's invention creates database proxy tables to represent methods of a remote service. In response to a database operation on the proxy table, the operation is automatically converted into the format for invoking a particular Web service using the mapping meta data which is stored in the database. In contrast, Brown relies on a mapping from an external DADX file and O'Farrell relies on a mapping maintained by a middle tier application server. Appellant's approach of maintaining the mapping information in the database provides for easier administration of the system and better security compared to reliance upon mappings from files or from application servers external to the database.

All told, although O'Farrell discusses mapping meta data, it is distinguishable from Appellant's claimed invention in several respects. First, O'Farrell's mapping is from data fields used to create a view on a client device to data fields of a back-end database, while Appellant's solution provides a proxy table which emulates a method of a remote Web service so as to enable the database system to automatically invoke a method the Web service in response to an operation on the proxy table. Additionally, Appellant's solution automatically creates the proxy table and mappings based on an interface definition of a Web service, while O'Farrell relies on a user to configure views for mapping enterprise data to client devices. Most significantly, O'Farrell does not use the mapping meta data for invoking a method of a Web service and converting the results into relational format as provided in Appellant's specification and claims.

Lastly, Appellant does not believe that the O'Farrell reference is properly considered as prior art as to Appellant's invention for the following reasons. The O'Farrell non-provisional patent application which was published as US PGPub 2005/0044164 has a filing date of December 23, 2003. This date is after the December 16, 2003 filing date of Appellant's non-provisional application. Moreover, Appellant's patent application claims the benefit of provisional application serial no. 60/320,009 (Docket No. SYB/0093.00) filed March 14, 2003. Although the O'Farrell reference also

claims priority to three provisional applications filed Dec. 23, 2002 (Serial No. 60/436,230), Jan. 23, 2003 (Ser. No. 60/442,810) and Apr. 7, 2003 (Ser. No. 60/461,588), Appellant's review of these three provisional applications finds that the disclosure included in the three provisional filings appears to be very different than that found in the published version of the O'Farrell application. Although the 35 U.S.C. 102(e) date of a reference may relate back to its earliest effective U.S. filing date, this requires that the prior provisional application(s) must properly support the subject matter used to make the rejection in compliance with 35 U.S.C. 112, first paragraph (See e.g., MPEP §706.02(f)(1)).

In response to Appellant's Amendment After Final which questioned whether prior provisional application(s) supported the referenced teachings, the Examiner indicated that the teachings of O'Farrell referenced in the Final rejection are supported in the "Hyperlinks on data" section at page 10 of O'Farrell Provisional Application 60/442,810 which reads as follows:

> An important function performed by the connector service is the mapping of data from third-party sources to client applications using metadata as the link. This is accomplished using Hyper-Data-Links. Hyper-Data-Links are similar to "Hyperlinks". The difference is where a hyperlink takes you from one web page to another, Hyper-Data-Links direct the Dextera Client to the specific "table" or "field" in a host application to retrieve the data desired.
> Hyper-Data-Links utilize the Dexterra Connector Web Service and ODBC physical connections to the host applications to provide a "link" or metapath between the data resident in the Enterprise Application and the target Dexterra Client. An important point here is that no enterprise application data is held in the Dexterra Server, thus assuring data consistency and integrity. Hyper-Data-Links provide the "path" to the specific data for the Dexterra Client application(s). This noninvasive approach to integration allows Dexterra to effectively connect to back-end systems with a minimum amount of disruption while maximizing data consistency and integrity.
>
> (O'Farrell Provisional Application 60/442,810, page 10, "Hyperlinks on data", emphasis added)

Although the above-referenced section of the provisional application does reference mapping of data third party sources to a client "using metadata as the link", it makes it even more clear that the mapping meta data described by O'Farrell is fundamentally different than Appellant's claimed invention. For one thing, the above text

from the provisional application makes no mention whatsoever of generating meta data or storing meta data in a database table. It also provides no teaching of using the stored meta data for converting a database operation into a format for invoking a method of a Web service. Instead, it describes Hyper-Data-Links which provide a client with a "path" or mapping to specifics tables or fields of a host application for retrieval of data. This is not comparable to Appellant's claimed invention.

3. Conclusion

As discussed in detail above, Brown and O'Farrell, either alone or in combination, do not include all the limitations of Appellant's claims 1-55. For the reasons stated, it is respectfully submitted that Appellant's claims 1-55 distinguish over the prior art and that the Examiner's rejection under Section 103 should not be sustained.

C. Conclusion

The present invention greatly improves the efficiency of the task of incorporating data available from a Web service into a database and performing relational operations over such data in a database system. It is respectfully submitted that the present invention, as set forth in the pending claims, sets forth a patentable advance over the art.

In view of the above, it is respectfully submitted that the Examiner's rejection of Appellant's claims under 35 U.S.C. Section 103 should not be sustained. If needed, Appellant's undersigned attorney can be reached at 925 465 0361. For the fee due for this Appeal Brief, please refer to the attached Fee Transmittal Sheet. This Appeal Brief is submitted electronically in support of Appellant's Appeal.

Respectfully submitted,

Date: November 14, 2007

/G. Mack Riddle/

G. Mack Riddle; Reg. No. 55,572
Attorney of Record

925 465 0361
925 465 8143 FAX

8. CLAIMS APPENDIX

1. A method for performing database operations on data obtained from a web service, the method comprising:

creating at least one proxy table in a database, each proxy table mapping to a method of the web service;

generating meta data about the mapping and storing the meta data in a database table of the database;

in response to a database operation on a particular proxy table, using the meta data for converting the database operation into a format for invoking a particular method of the web service based upon the corresponding mapping;

invoking the particular method of the web service;

converting results obtained from invoking the particular method into data for use at the database based upon the corresponding mapping;

performing the database operation on the data at the database to generate a result set; and

returning the result set in response to the database operation.

2. The method of claim 1, wherein the web service comprises a service remotely available via a network.

3. The method of claim 1, wherein the web service has a Web Services Description Language (WSDL) interface.

4. The method of claim 3, wherein said creating step includes creating said at least one proxy table based upon the WSDL interface.

5. The method of claim 3, wherein said creating step includes substeps of:

obtaining the WSDL interface from the web service; and

creating said at least one proxy table based upon the WSDL interface.

6. The method of claim 1, wherein said generating step includes creating meta data identifying a particular method of the web service to be invoked when a database operation is received on a particular proxy table.

7. The method of claim 1, wherein said creating step includes mapping arguments of the method to fields of the proxy table.

8. The method of claim 1, wherein said creating step includes mapping arguments of the method to equivalent database data types.

9. The method of claim 1, wherein said creating step includes creating an object encapsulating the mapping of a web method to the database.

10. The method of claim 1, wherein said generating step includes storing meta data about the mapping between said at least one proxy table and methods of the web service in a system table of the database.

11. The method of claim 10, wherein said step of converting results includes consulting the mapping for converting the results into data for application at the database.

12. The method of claim 1, wherein the database operation includes a JOIN operation and said step of performing the database operation includes joining data obtained from invoking the particular method of the web service with data stored in the database in generating the result set.

13. The method of claim 1, wherein said step of converting the database operation includes binding data from the database operation to a Simple Object Access Protocol (SOAP) call for invoking the particular method of the web service.

14. The method of claim 1, wherein said step of converting the database operation includes converting data from the database operation into Extensible Markup

Language (XML) format.

15. The method of claim 1, wherein said step of converting the database operation includes creating a Simple Object Access Protocol (SOAP) request for invoking the particular method of the web service.

16. The method of claim 15, wherein said step of invoking the particular method includes transmitting the SOAP request to a remote web service.

17. The method of claim 1, wherein said step of invoking the particular method includes receiving results from the web service.

18. The method of claim 1, wherein said step of converting results includes converting results received in Simple Object Access Protocol (SOAP) format.

19. The method of claim 1, wherein said step of converting results includes converting results received in Extensible Markup Language (XML) format.

20. A computer-readable medium having processor-executable instructions for performing the method of claim 1.

21. A downloadable set of processor-executable instructions for performing the method of claim 1 stored on a web server.

22. In a computer connected to a network and having access to a remote service, a system for performing operations at a database on data obtained from the remote service, the system comprising:

a mapping module for creating database tables representing at least some methods of the remote service accessed through a defined interface and storing mapping data regarding methods of the remote service in a database system table;

an invocation module for converting a database operation on a database table

representing a method of the remote service into a call for invoking the method using the mapping data;

a communication module for transmitting the call for invoking the method to the remote service, and returning result values from invoking the method to the database; and

a conversion module for converting result values received from the method into database format, performing the database operation on the converted result values to generate a database result set, and returning the database result set in response to the database operation.

23. The system of claim 22, wherein the remote service comprises an application available via a network.

24. The system of claim 22, wherein the defined interface comprises a Web Services Description Language (WSDL) interface.

25. The system of claim 24, wherein said mapping module creates the database tables based on the WSDL interface.

26. The system of claim 22, wherein said mapping module creates mapping data identifying a particular method of the remote service to be invoked when an operation is received on a given database table.

27. The system of claim 22, wherein said mapping module maps arguments of a method to columns of a database table.

28. The system of claim 22, wherein each database table created by the mapping module represents a method of the remote service.

29. The system of claim 22, wherein said mapping module creates an object encapsulating the mapping of a method of the remote service to a database table.

30. The system of claim 22, further comprising:

a mapping repository for storing mapping data regarding mappings between database tables and methods of the remote service in the database system table.

31. The system of claim 30, wherein the conversion module consults the mapping repository for converting result values into database format.

32. The system of claim 22, wherein the operation received on the database table comprises a JOIN operation and said conversion module joins result values obtained from invoking the method with data stored in the database.

33. The system of claim 22, wherein said invocation module binds the data from the operation to a Simple Object Access Protocol (SOAP) call for invoking the method of the remote service.

34. The system of claim 22, wherein said invocation module converts data from the database operation into Extensible Markup Language (XML) format.

35. The system of claim 22, wherein said invocation module creates a Simple Object Access Protocol (SOAP) request for invoking the method of the remote service.

36. The system of claim 35, wherein said communication module sends the SOAP request to the remote service.

37. The system of claim 22, wherein said conversion module converts result values received in Simple Object Access Protocol (SOAP) format into database data types.

38. The system of claim 22, wherein said conversion module converts result values received in Extensible Markup Language (XML) format into database data types.

39. The system of claim 22, wherein said database operation received by the invocation module comprises a database query received from a user and said conversion module returns a database result set to the user in response to said database query.

40. In a database system, a method for performing database queries on data available from an application, the method comprising:

establishing communication between a database and an application having an interface;

creating database tables to represent at least some functions of the application based on the interface, each database table mapping to a corresponding function of the application;

generating meta data about the mapping and storing the meta data in a system table of the database;

in response to a database query received on a database table corresponding to a function of the application, generating input arguments expected by the function based on the database query and the mapping meta data;

invoking the function with the input arguments and receiving results from invoking the function;

converting the results into a database result set; and

returning the database result set in response to the database query.

41. The method of claim 40, wherein the application comprises a web service.

42. The method of claim 40, wherein the application comprises a service available via a network.

43. The method of claim 40, wherein the interface comprises a Web Services Description Language (WSDL) interface.

44. The method of claim 40, wherein said generating step o includes creating meta data identifying a particular function to be invoked when an operation is received on

a given database table.

45. The method of claim 40, wherein said step of creating database tables includes mapping arguments of a given function to columns of the corresponding database table.

46. The method of claim 40, wherein said step of invoking the function includes binding data from the database query to a Simple Object Access Protocol (SOAP) call.

47. The method of claim 40, wherein said step of invoking the function includes converting data from the database query into Extensible Markup Language (XML) format.

48. The method of claim 40, wherein said step of invoking the function includes creating a Simple Object Access Protocol (SOAP) request for invoking the function.

49. The method of claim 48, wherein said step of invoking the function includes transmitting the SOAP request to a remote server.

50. The method of claim 40, wherein said step of invoking the function includes receiving results in Extensible Markup Language (XML) format.

51. The method of claim 40, wherein said step of invoking the function includes receiving results in Simple Object Access Protocol (SOAP) format.

52. The method of claim 40, wherein said step of converting the results includes converting results received in Simple Object Access Protocol (SOAP) format.

53. The method of claim 40, wherein said step of converting the results includes converting results received in Extensible Markup Language (XML) format.

54. A computer-readable medium having processor-executable instructions for performing the method of claim 40.

55. A downloadable set of processor-executable instructions for performing the method of claim 40 stored on a web server.

9. EVIDENCE APPENDIX

This Appeal Brief is not accompanied by an evidence submission under §§ 1.130, 1.131, or 1.132.

10. RELATED PROCEEDINGS APPENDIX

Pursuant to Appellant's statement under Section 2, this Appeal Brief is not accompanied by any copies of decisions.